```
let S = \langle x \ y \ z.x \ z \ (y \ z)
let K = \xy.x
S K K
 <whitespace>
\rightarrow " " <whitespace opt>
 \rightarrow " "
 <line_end>
 \rightarrow <whitespace opt> ("\n" | "\0")
\rightarrow ("\n" | "\0")
 <variable>
\rightarrow <letter> <variable_opt>
\rightarrow < letter>
\rightarrow ("S" | "K" | "x" | "y" | "z")
 <variables>
\rightarrow <variable> <variables opt>
\rightarrow <\!\! \text{variable} \!\! > <\!\! \text{whitespace} \!\! > <\!\! \text{variable} \!\! > <\!\! \text{variables\_opt} \!\! >
\rightarrow <variable> <whitespace> <variable> <whitespace> <variable> <v
ables opt>
 → <variable> <whitespace> <variable> <whitespace> <variable>
 \rightarrow "x y z"
 <expression>
 \rightarrow <expression opt> <line end>
 \rightarrow <definition> <line end>
\rightarrow <declaration> <whitespace> "=" <whitespace> <term> <line end>
<line_end>
\rightarrow "let S = " <term> "\n"
\rightarrow "let S = " <abstraction> "\n"
\rightarrow "let S = \" <
variables> "." <
term> "\n"
\rightarrow "let S = \x y z." <term> "\n"
```

```
\rightarrow "let S = \x y z." <application> "\n"
\rightarrow "let S = \x y z." < operand > < application opt > "\n"
\rightarrow "let S = \x y z." < variable > < application opt > "\n"
\rightarrow "let S = \xy z." < variable > < whitespace > < operand > < application opt >
"\n"
\rightarrow "let S = \xy z.x " < operand > < application opt > "\n"
\rightarrow "let S = \x y z.x " < variable > < application opt > "\n"
\rightarrow "let S = \x y z.x " < variable > < whitespace > < operand > < applica-
tion opt> "\n"
\rightarrow "let S = \xy z.x " \x
\rightarrow "let S = \xy z.x z" < operand > "\n"
\rightarrow "let S = \xy z.x z" < term par> "\n"
\rightarrow "let S = \xy z.x z (" < term > ")\n"
\rightarrow "let S = \x y z.x z (" <application> ")\n"
\rightarrow "let S = \xy z.x z (" < operand > < application opt > ")\n"
\rightarrow "let S = \xspace x y z.x z (" < operand > < whitespace > < operand > < applica-
tion opt>")\n"
\rightarrow "let S = \xy z.x z (" < operand > < whitespace > < operand > ")\n"
\rightarrow "let S = \x y z.x z (" < variable > < whitespace > < variable > ")\n"
\rightarrow "let S = \xy z.x z (y z)\n"
program>
\rightarrow <expression>                                                                                                                                                                                                                                                                                                                                                   
\rightarrow <expression> <expression>                                                                                                                                                                                                                                                                                                                                                
→ <expression> <expression>                                                                                                                                                                                                                                                                                                                                                 <
\rightarrow <expression> <expression>
\rightarrow <expression opt> end> <expression> <expression>
\rightarrow <definition> end> <expression> <expression>
\rightarrow <definition> end> <expression opt> end> <expression>
\rightarrow <definition> end> <definition> <lie end> <expression>
\rightarrow <definition> end> <definition> end> <expression opt>
end>
\rightarrow <definition> end> <definition> end> <term> end>
"let S = \langle x \ y \ z . x \ z \ (y \ z) \langle n \rangle
let K = \langle x y.x \rangle n
S K K \setminus 0"
```